

# DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO DE RUTEO PARA VEHÍCULOS ELÉCTRICOS

Felipe Olaya  
Universidad Eafit  
Colombia  
folayao@eafit.edu.co

Laura Sánchez  
Universidad Eafit  
Colombia  
lsanchezc@eafit.edu.co

Manuela Valencia  
Universidad Eafit  
Colombia  
mvaleciat@eafit.edu.co

Mauricio Toro  
Universidad Eafit  
Colombia  
mtorobe@eafit.edu.c

o

## RESUMEN

Ante la creciente problemática ambiental de la cual una de sus causas es el constante uso de tecnologías emisoras de gases efecto invernadero como son los autos a combustible, se hace necesario buscar alternativas viables y más amigables con el medio ambiente.

Partiendo de este problema se busca implementar una solución a partir del uso de autos eléctricos. Sin embargo, estos tienen varias restricciones como el rango de conducción y el extenso tiempo de carga, por lo que se necesita hallar una solución que permita buscar la ruta más óptima (teniendo en cuenta el tiempo empleado en los viajes y cargas) para llevar a cabo los recorridos. De esta manera se pretende crear una solución cimentada en el diseño de algoritmos y estructuras de datos para resolver el problema, tomando como base otros trabajos relacionados y a partir de estos llegar a una solución propia, viable y rápida.

## PALABRAS CLAVE

Diseño de algoritmos, estructuras de datos, grafos, ruteo de vehículos, soluciones óptimas, análisis de datos, TSP.

## PALABRAS CLAVE DE LA CLASIFICACIÓN DE LA ACM

Software and its engineering → Software creation and management → Designing software → Software implementation planning → Software design techniques

## 1. INTRODUCCIÓN

Actualmente son muchas las personas que usan tecnologías y artefactos dependientes del petróleo, y emisores de diferentes gases de efecto invernadero. Esto ha estado ocasionando un desgaste cada vez peor en el planeta, lo cual ha sido altamente notable en los últimos años con el descontrol climático.

Por esto, es importante actuar lo más pronto posible, pues es un problema que involucra la vida y la salud de todas las personas, y si no se actúa rápido, llegaremos a un punto en el que ya no haya nada que hacer.

Una manera de disminuir esta problemática es con el uso de vehículos eléctricos, pero tienen cierta limitación, y es el bajo rango de conducción y el alto tiempo de carga que

requiere. A esto, es a lo que se quiere dar solución con este proyecto.

Tal y como ésta, existen muchas otras problemáticas relacionadas, tal y como el problema del agente viajero, el problema de Enrutamiento de Vehículos eléctricos de Carga parcial con Carga lineal, y otros más que serán explicados a continuación. Mediante el estudio de éstas se comenzará a dar resolución a nuestro problema de Enrutamiento de Vehículos Eléctricos.

## 2. PROBLEMA

Se requiere diseñar un algoritmo que permita obtener al usuario la ruta más óptima para que una flota de camiones (cuyo número no es fijo) para visitar un conjunto de clientes. La ruta obtenida deberá ser aquella que minimice lo mejor posible el tiempo total del recorrido con carga lineal y parcial (suma del tiempo de carga con el de visita a los clientes).

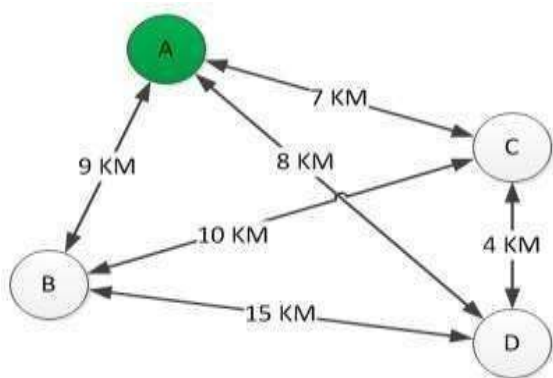
## 3. TRABAJOS RELACIONADOS

Se va a hacer alusión a los siguientes problemas: Problema del Agente Viajero, Problema de Enrutamiento de Vehículos Eléctricos de Carga parcial con Carga lineal, Problema de enrutamiento de vehículos eléctricos con tiempo de carga y tiempo de viaje variable, Problema de enrutamiento de vehículos

Estos problemas mencionados tienen una relación con nuestro problema a tratar, las diferencias que manejan consisten en el enfoque como tal cada uno de los problemas, por ejemplo, en la sección 3.3 se enfoca en los tiempos de carga, el 3.1 en el costo, y así varía dependiendo del contexto en el que se esté hablando.

### 3.1 Problema del Agente Viajero

Se tiene una cantidad de puntos y el costo de viajar entre cada par de ellos, para eso se usa el TSP, es encontrar la forma más barata de visitar todos los puntos y regresar a su punto de partida. En la versión estándar, los costos de viaje son simétricos en el sentido de que viajar del punto X al punto Y cuesta tanto como viajar de Y a X. En la gráfica 1 podemos ver un ejemplo de costos asimétricos entre los puntos. [1]



**Gráfica 1:** en este caso los costos se miden en distancia. [2]

### 3.2 Problema de Enrutamiento de Vehículos Eléctricos de Carga parcial con Carga lineal

Los vehículos eléctricos son una de las tecnologías más prometedoras para reducir la dependencia del petróleo y las emisiones de gases de efecto invernadero. El uso de vehículos eléctricos en el transporte de mercancías y pasajeros introduce una nueva familia de problemas de enrutamiento de vehículos (VRP), los llamados VRP eléctricos (eVRP). Como su nombre lo indica, los eVRP extienden los VRP clásicos para dar cuenta de las características técnicas de los vehículos eléctricos donde hay que tener en cuenta el tiempo y costo de carga. Sin embargo, se ha visto que tratar la carga de manera no lineal da un costo extra al calcular las rutas, por lo que a la hora de buscar solución al clásico problema VRP se hace más factible tomar la carga como una función lineal [3].

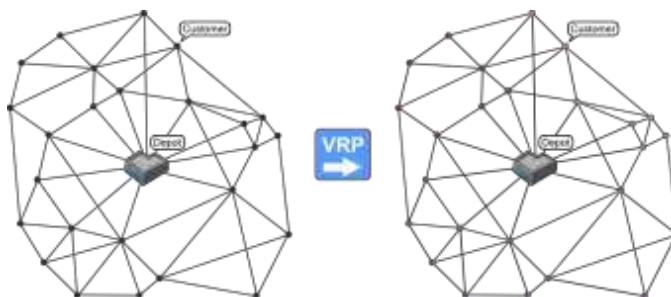
### 3.3 Problema de enrutamiento de vehículos eléctricos con tiempo de carga y tiempo de viaje variable

Se tiene enrutamiento de vehículos eléctricos con tiempo de carga y tiempo de viaje variable para abordar algunos problemas operativos tales como la limitación del alcance y la demanda de carga. El modelo se resuelve mediante el uso de un algoritmo genético para obtener las rutas, la hora de salida del vehículo en el depósito y el plan de carga. [4]

### 3.4 El problema de enrutamiento de vehículos

El problema de enrutamiento de vehículos (VRP) es un nombre genérico dado a toda una clase de problemas en los que se debe determinar un conjunto de rutas para una flota de vehículos en uno o varios depósitos para una cantidad de ciudades o clientes geográficamente dispersos. El objetivo del VRP es entregar un conjunto de clientes con demandas conocidas sobre rutas de vehículos de costo mínimo que se originan y terminan en un depósito [5]. En el gráfico 2 se

puede ver una imagen de una entrada típica para un problema de VRP y una de sus posibles salidas:



**Gráfica 2**

## 4. TSP (Problema del Agente Viajero)

El problema del agente viajero tiene como objetivo, juntar todos los nodos de una red, visitándolos solo una vez y regresando al punto de partida; también como ideal de este algoritmo es minimizar costos (distancia total recorrida entre, nodo y nodo).

### 4.2 Criterios de diseño de la estructura de datos

Usamos el agente viajero, por los siguientes motivos: Para poder tener una mayor eficiencia en el tiempo, ya que hallando la distancia más corta con éstas poder hacer las mediciones de tiempos de forma ágil garantizando una buena solución, con la memoria decidimos usar una matriz de adyacencia para poder representar grafos, a pesar de que esta es más pesada, es más eficiente a la hora de ir a recuperar la información para ir de un nodo a otro, siendo así más eficiente con los tiempos.

### 4.3 Cálculo de la complejidad del algoritmo

Método	AgenteViajero	
Complejidad	$O(n)=n^2$	$O(n)=n*m$

Siendo  $n$  el número de nodos existentes en el grafo (puntos en el mapa: clientes, estaciones, depósito) y la variable  $m$  los nodos en evaluación, es decir los sucesores por los que sigue la ruta.

**4.7 Criterios de diseño del algoritmo** Se decidió hacer este algoritmo de esta forma, porque a la hora de llevar a cabo la lectura de los datos, resulta mucho más fácil su almacenamiento y posterior manejo, se entiende el problema como la representación abstracta de un grafo con las coordenadas estipuladas en los ficheros.

Estas coordenadas representan las ubicaciones de las paradas existentes (ya sea depósito, cliente o estación de carga), la representación del grafo se lleva a cabo mediante una matriz de adyacencia que si bien ocupa una buena cantidad de espacio, el problema se diseña para un grafo conexo, lo que implica que no se perderá nada del espacio apartado, por otra

parte; manejar los datos con ésta se hace mucho más sencillo, pues tanto almacenar una distancia, como retornarla tiene una complejidad  $O(1)$ .

Para el desarrollo de la búsqueda de una ruta óptima se emplea actualmente una implementación del algoritmo de El Vecino más Cercano, conocido como la solución al problema del Agente Viajero. esta implementación nos permite Hallar una ruta óptima teniendo en cuenta los costos (distancias existentes entre los puntos), se mira cual es el menor partiendo del depósito para llegar a los clientes y pasando en ocasiones por las estaciones de carga

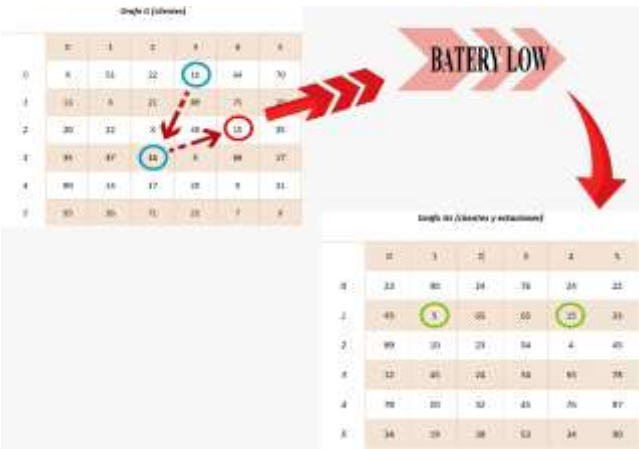
5. ALGORITMO PARA RUTEO DE VEHÍCULOS ELÉCTRICOS

A continuación, explicamos la estructura de datos y el algoritmo.

5.1 Estructura de datos

Grafos como matrices de adyacencia.

5.2 Operaciones de la estructura de datos



**Gráfica 3:** Búsqueda del vecino más cercano por medio de las distancias entre cada vértice, primero se hace con los clientes, y si la batería no es suficiente para el siguiente nodo, se va al grafo de clientes y estaciones, buscando cuál estación es más cercana y luego busca la distancia desde ésta al vértice previsto.

*Se anexa a la entrega para que se pueda visualizar mejor*

5.3 Criterios de diseño de la estructura de datos

En cuanto a al grafo representado por una matriz, a pesar de que ocupa un gran espacio, teniendo en cuenta que cada grafo es conexo, garantiza que ningún espacio de la matriz será desaprovechado, aparte de esto, nos permite realizar búsquedas e insertar fácilmente información en este, con un bajo nivel de complejidad.

La representación de datos en forma de arreglo permite obtener y almacenar datos de con una maja complejidad.

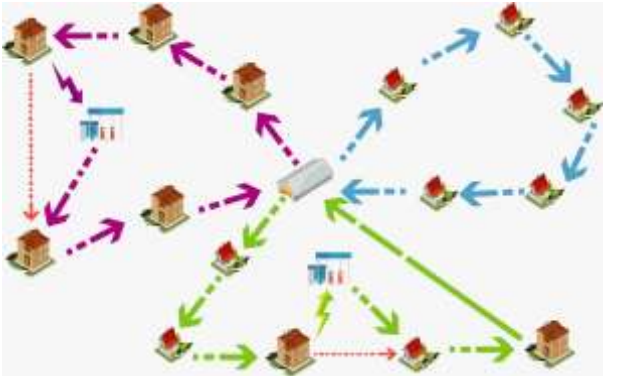
Las listas y pares usadas como estructuras auxiliares permiten guardar información de manera práctica, ahorrándonos ciertos procesos de más, y creación de nuevas variables.

5.4 Análisis de Complejidad

Manejo de datos con una matriz	$O(1)$
Manejo de datos con un arreglo	$O(1)$
Manejo de datos con listas y pares	$O(1)$

Siendo n el número de nodos existentes en el grafo (puntos en el mapa: clientes, estaciones, depósito) y la variable m los nodos en evaluación, es decir los sucesores por los que sigue la ruta.

5.5 Algoritmo



**Gráfica 4:** Paso a paso cómo se calculan las rutas para los vehículos según tiempo, distancia y energía.

5.6 Cálculo de la complejidad del algoritmo

En cada uno de los casos, es decir, tanto en el mejor como en el peor, la complejidad de éste depende directamente de la cantidad de nodos que se crean en el grafo una vez leído el archivo, aparte de esto, la complejidad aumenta mientras más distante estén los clientes, debido a que la batería se agotará después de visitar un pequeño número de clientes, y por esto, serán más las estaciones de carga que se tendrá que visitar.

Para este algoritmo, el mejor de los casos sucederá cuando el camión logre visitar la totalidad de clientes sin tener que desplazarse a una estación de carga, y el peor de los casos sucede cuando es necesario visitar una gran

cantidad de estaciones de carga. Teniendo en cuenta estos dos casos, el caso promedio ocurre cuando hay distancias moderadas entre los clientes.

Sub problema	Complejidad
Leer el archivo y almacenarlo en grafos	$O(N*M)$
Calcular cada uno de los desplazamientos	$O(M*N)$
<b>Complejidad Total</b>	$O(M*N)$

## 5.7 Criterios de diseño del algoritmo

Entre las principales razones que nos llevaron a escoger este diseño de algoritmo, está su eficiencia, ya que al tomar siempre la ruta más corta no se hace necesario tener que devolverse a mirar cada uno de los posibles caminos. Después de probar diferentes diseños, se pudo notar que este tiene mejor rendimiento, arroja resultados en más poco tiempo, y por último, aunque su complejidad podría mejorarse, la que tiene actualmente  $O(m*n)$  permite que este se ejecute rápidamente con los casos propuestos, .

## 5.8 Tiempos de Ejecución

Tiempos tomados en la segunda entrega para cada método construido hasta el momento

Tiempos tomados en la entrega final para cada método:

Archivo .txt	time-Agente viajero	time-Leer
tc2c320s38ct1	145	76
tc2c320s24cf0	103	139
tc2c320s24cf1	118	62
tc2c320s24cf4	63	167
tc2c320s24ct4	59	178
tc2c320s38cf0	62	70
tc2c320s38cf1	64	65
tc2c320s38cf4	109	63
tc2c320s38ct0	66	68
tc2c320s38ct1	65	78
tc2c320s38ct4	66	74

## Uso de memoria del algoritmo final:

Archivo txt	B
tc2c320s24cf0	375.856.000
tc2c320s24cf1	375.856.000
tc2c320s24cf4	375.856.000
tc2c320s24ct0	375.856.000
tc2c320s24ct1	375.856.000
tc2c320s24ct4	375.856.000
tc2c320s38cf0	375.856.000
tc2c320s38cf1	375.856.000
tc2c320s38ct4	375.856.000

## CONCLUSIONES

El número de rutas generadas a partir del algoritmo depende no solo de la cantidad de nodos existentes, sino también de la distancia entre estos pues de ahí depende, dado el tiempo y la energía que se toma cada camión, las desviaciones y las rutas que se deben seguir para cumplir con la visita a todos los clientes.

Para el diseño del algoritmo, se busca no solo soluciones viables, también se busca que se ejecute de manera rápida por lo que se buscaron estructuras que, si bien ocupan una buena capacidad en memoria, garantizan un buen funcionamiento.

## AGRADECIMIENTOS

Queremos agradecer especialmente al monitor de la materia estructura de datos y algoritmos 2, Mateo Agudelo, pues sin su implementación de la clase Pair, hubiese sido más difícil el diseño de nuestra solución al problema planteado.

Agradecemos igualmente a la Universidad Eafit y al Departamento de Informática que con sus recursos y oportunidades de aprendizaje permiten llevar a cabo estos proyectos.

## REFERENCIA

1. Travelling Salesman Problem | Set 1 (Naive and Dynamic Programming). <https://www.geeksforgeeks.org/travelling-salesman-problem-set-1/>
2. Salazar, B. PROBLEMA DEL AGENTE VIAJERO - TSP. <https://www.ingenieriaindustrialonline.com/herramientas-para-el-ingeniero-industrial/investigacion/C3%B3n-de-operaciones/problema-del-agente-viajero-tsp/>
3. Montoya, A. , Guéret, C. , Mendoza J. and Villegas J. The electric vehicle routing problem with partial charging and nonlinear charging function. <https://hal.archives-ouvertes.fr/hal-01245232/document>

4. Shao, S. , Guan W. , He, Z. and Bi, J. Electric Vehicle Routing Problem with Charging Time and Variable Travel Time. Mathematical Problems in Engineering Volume 2017 (2017), Article ID 5098183, 13 pages <https://doi.org/10.1155/2017/5098183>.

5. Vehicle Routing Problem.

## **ANEXO TRABAJO EN EQUIPO**

### **Entrega 1 del proyecto:**

Para esta primera entrega tratamos de repartirnos el trabajo de manera equitativa, de modo que todos trabajáramos por igual.

Por un lado, Laura se encargó de conseguir información acerca de problemas similares al propuesto, para lo cual uso el internet, y el conocimiento de personas cercanas.

Por otro lado, Felipe se encargó de la definición del problema, y elaboración del primer Adelanto del resumen del proyecto.

Por último, Manuela, se encargó del diseño de las gráficas correspondientes y de la redacción del problema.

### **Entrega 2 del proyecto:**

En esta segunda entrega, Laura y Manuela se encargaron del primer Adelanto del diseño del algoritmo con el que se buscó dar solución a la problemática, así como de la toma de tiempos para cada método, a su vez, Felipe se encargó de la redacción del documento a entregar, con todos sus requerimientos.

### **Entrega final del proyecto:**

Para esta entrega, Laura se encargó principalmente de perfeccionar el funcionamiento del método leer y la adaptación del algoritmo agente viajero para este proyecto, y corrección de errores que impedían el buen funcionamiento del algoritmo.

Por su lado, Manuela se encargó del mejoramiento del método agente viajero, adaptación de las clases encargadas de crear el grafo después de leer el archivo y corrección de errores que impedían el buen funcionamiento del algoritmo.

Felipe se encargó de aplicar el formato estipulado para la salida de los datos, y de verificar los resultados arrojados con cada archivo.

En esta última entrega entre todos se desarrolló el informe propuesto, tanto de la adición de nueva información, como de la corrección de la información escrita anteriormente.

